Hybridization methods for the analysis of nonlinear systems

Eugene Asarin · Thao Dang · Antoine Girard

Abstract   In this article, we describe some recent results on the hybridization methods for the analysis of nonlinear systems. The main idea of our hybridization approach is to apply the hybrid systems methodology as a systematic approximation method. More concretely, we partition the state space of a complex system into regions that only intersect on their boundaries, and then approximate its dynamics in each region by a simpler one. Then, the resulting hybrid system, which we call a hybridization, is used to yield approximate analysis results for the original system. We also prove important properties of the hybridization, and propose two effective hybridization construction methods, which allow approximating the original nonlinear system with a good convergence rate.

1 Introduction

Hybrid systems, that is, systems exhibiting both continuous and discrete dynamics, have proven to be a useful mathematical model for various physical phenomena and engineering systems. One typical example is a chemical batch plant where a computer is used to supervise complex sequences of chemical reactions, each of which is modeled as a continuous process. In addition to discontinuities introduced by the computer, most physical processes admit components (e.g. switches, valves) and phenomena (e.g. collision, emptying of tanks) whose most useful models are discrete.

E. Asarin
Université Paris 7, LIAFA, 2 pl. Jussieu, 75251 Paris, Cedex 5, France
e-mail: Eugene.Asarin@liafa.jussieu.fr

T. Dang (✉)
VERIMAG, 2 ave. de Vignate, 38610 Gieres, France
e-mail: Thao.Dang@imag.fr

A. Girard
Université Joseph Fourier, LMC, B.P. 53, 38041 Grenoble Cedex 9, France
e-mail: Antoine.Girard@imag.fr

Hybrid system models arise in many applications, such as chemical process control, avionics, robotics, automobiles, and most recently in molecular biology.

Due to the safety critical features of many such applications, formal analysis is a topic of particular interest. The goal of *formal verification* is to prove that the (designed) system satisfies a property, and the goal of *controller synthesis* is to control the system (in other words to design a controller) so that the system satisfies a desired specification. Due to the complexity and scale of real-life applications, automatic analysis is very desirable. This is a motivation to adopt the algorithmic approach which consists in building software tools that can analyze automatically all the behaviors of a given system. Although the research on the algorithmic analysis of hybrid systems has achieved considerable results in the development of theoretical foundations and tools, their applications to real-life problems are still limited. A major component in any algorithm for analyzing hybrid systems is an efficient method to handle their continuous dynamics described by differential equations (since their discrete dynamics can be handled using well-developed verification methods in computer science). While many well-known properties of affine or piecewise affine systems and other simpler systems (such as systems with piecewise constant derivatives) can be exploited to design relatively efficient methods, nonlinear systems are much more difficult to analyze.

In this paper, our object of study is a complex system (that could be hybrid or not), and we apply the hybrid systems methodology as a systematic approximation method. More concretely, we propose an approach to study a complex system (with nonlinear dynamics for example) by approximating it with a simpler system, for which well-developed analysis tools exist. To this end, we partition the state space of the system into regions that only intersect on their boundaries and then approximate locally its dynamics in each region by a simpler dynamics. Globally, the dynamics of the approximate system changes when moving from one region to another. Due to these switchings, the approximate system behaves like a hybrid system and we thus call this approximation process *hybridization*. Then, the resulting system is used to yield approximate analysis results for the original system. The usefulness of this approach (in terms of accuracy and computational tractability) depends on the choice of the approximate system. We consider two classes of approximate systems: piecewise affine and piecewise multi-affine. We show that the use of these classes allows approximating the original nonlinear system with a good convergence rate.

The essence behind the hybridization approach is not new. However, the novelty in our work is that our approximation method is "systematic" in the sense that given a complex system, the method can automatically compute a system which approximates the original system with a guaranteed error bound and thus preserves the properties of interest. In addition, the approximate system can be studied using available tools for the formal analysis of hybrid systems.

The paper is organized as follows. In Sect. 2, we discuss the model we use for describing hybrid systems. We then discuss some common properties of hybrid systems and briefly review the existing results on their algorithmic analysis. This section includes the basic definitions and notations neccessary for subsequent discussions. In Sect. 3, we describe the main principles of our hybridization approach. In Sect. 4, we prove important properties of the method and present a comparison of our method with previous results. In Sect. 5, we show two effective methods for constructing hybridizations: one produces affine hybridizations and the other produces multi-affine hybridizations. The last section contains some examples illustrating our approach.

2 Hybrid systems framework

2.1 Hybrid model

A hybrid system is a system whose evolution consists of successions of continuous phases and discrete events. Various hybrid systems models have been proposed and this remains an active research area [14,37]. In this paper, we use the following model, which is a variant of the hybrid automaton proposed in [1]. The reason we choose this model is that it can capture naturally a wide range of hybrid behaviors and, moreover, provides a framework suitable for the problems we tackle in this work.

Definition 1 A *hybrid system* $H = (L, n, p, E, F, Inv, G, R)$ consists of:

- A set of discrete locations L.
- An integer n, the dimension of the continous state space.
- An integer p, the dimension of the continuous input.
- A set of discrete transitions $E \subseteq L \times L$.
- A collection of continuous vector fields $F = \{F_l | \ l \in L\}$. For each location $l \in L$, $F_l = (U_l, f_l)$ where $U_l \subset \mathbb{R}^p$ is a set of inputs and $f_l : \mathbb{R}^n \times U_l \to \mathbb{R}^n$. We assume that the vector fields f_l are Lipschitz continuous.[1] The admissible input functions are piecewise continuous.
- A collection of invariants $Inv = \{Inv_l | \ l \in L\}$. For each location $l \in L$, $Inv_l \subseteq \mathbb{R}^n$.
- A collection of guards $G = \{G_e | \ e \in E\}$. For each discrete transition $e = (l, l') \in E$, $G_e \subseteq Inv_l$.
- A collection of reset relations $R = \{R_e | \ e \in E\}$. For each discrete transition $e = (l, l') \in E$, $R_e \subseteq G_e \times Inv_{l'}$.

The hybrid state space of H is $\mathcal{S} = \bigcup_{l \in L}\{l\} \times Inv_l$. Then, the state of the hybrid system is a pair (l, x) where the discrete state is $l \in L$ and the continuous state is $x \in Inv_l$. A hybrid system can be thought as the interaction between a discrete and a continuous process. The discrete part of the system is described by an automaton (L, E) whose transitions are triggered when the continuous variable reaches the associated guard. Between two transitions, the continuous process evolves according to the continuous vector field associated with the active discrete state. In the following, we define formally the notion of traces of a hybrid system.

Definition 2 Let L be a set of discrete locations and $n \in \mathbb{N}$, a *hybrid trajectory* on $L \times \mathbb{R}^n$ is a triple $(\mathcal{I}, \mathcal{Q}, \mathcal{X})$ where

- $\mathcal{I} = \{I_k | \ 0 \le k \le N\}$ is a sequence of intervals (we can have $N = +\infty$) such that
    - If $N = +\infty$, then for all $k \in \mathbb{N}$, $I_k = [t_k, t'_k]$ with $t'_k = t_{k+1}$.
    - If $N < +\infty$, then $I_N = [t_N, t'_N]$ or $I_N = [t_N, +\infty)$ and for all $0 \le k \le N - 1$, $I_k = [t_k, t'_k]$ with $t'_k = t_{k+1}$.
    - In both cases, the initial time is $t_0 = 0$.
- $\mathcal{Q} = \{q_k | \ 0 \le k \le N\}$ is a sequence of locations.
- $\mathcal{X} = \{x_k | \ 0 \le k \le N\}$ is a sequence of continuous, piecewise differentiable functions. For all $0 \le k \le N$, $x_k : I_k \to \mathbb{R}^n$. The dotted function $\dot{x}_k$ denotes the derivative of x_k.

---

[1] This requirement is needed for existence of continuous trajectories as well as for error estimation.

If $N < +\infty$ and $I_N = [t_N, t'_N]$ then the hybrid trajectory is said to be *finite*. If $N = +\infty$ and $\lim_{k \to +\infty} t_k < +\infty$ then the hybrid trajectory is said to be *Zeno*. Otherwise, it is said to be *infinite*.

It should be noted that for simplicity, the functions x_k are assumed to be continuous and piecewise differentiable. However, it is possible to consider larger and thus more expressive classes of functions, such as absolutely continuous functions (see for instance [11]).

Intuitively, the intervals I_k represent the time intervals where the hybrid system evolves continuously according to a differential equation satisfied by the function x_k. The time point t'_k is the instant where a discrete transition occurs to switch from the location q_k to the location q_{k+1}. More formally, the evolution of a hybrid system is described by the notion of *hybrid traces*.

Definition 3 Let $H = (L, n, p, E, F, Inv, G, R)$, a hybrid trajectory $\tau_h = (\mathcal{I}, \mathcal{Q}, \mathcal{X})$ on $L \times \mathbb{R}^n$ is a *hybrid trace* of the hybrid system H if the following conditions hold:

- **continuous evolution:** For all $0 \le k \le N$, there exists a piecewise continuous input u_k such that for all $t \in I_k$, $x_k(t) \in Inv_{q_k}$, $u_k(t) \in U_{q_k}$ and at each time t where u_k is continuous:

$$\dot{x}_k(t) = f_{q_k}(x_k(t), u_k(t)).$$

- **discrete evolution:** For all $0 \le k \le N-1$, $e_k = (q_k, q_{k+1}) \in E$, $x_k(t'_k) \in G_{e_k}$ and $(x_k(t'_k), x_{k+1}(t_{k+1})) \in R_{e_k}$.

The set of hybrid traces of H is denoted by $\mathcal{T}_h(H)$. The subset of $\mathcal{T}_h(H)$ consisting of the infinite hybrid trajectories is denoted by $\mathcal{T}_h^\infty(H)$.

We remark that the continuous dynamics of a hybrid system are described by differential equations with inputs which might not be control inputs, and in this case they must be thought as disturbances or uncertainties. Hence, a hybrid system is in general non-deterministic (though non-stochastic) in the sense of admitting more than one trajectory (in fact, uncountably many trajectories) from a given initial state. In this paper, we are mainly interested in the evolution of the continuous state of the hybrid system. We thus introduce the following notion of *continuous traces*.

Definition 4 A *continuous trace* of the hybrid system H is a pair $\tau_c = (I, x)$ consisting of an interval I and a piecewise C^1 function[2] $x : I \to \mathbb{R}^n$ such that there exists at hybrid trace $(\mathcal{I}, \mathcal{Q}, \mathcal{X})$ of H satisfying:

- $I = \bigcup_{k=0}^{k=N} I_k$.
- For all $0 \le k \le N$, for all $t \in (t_k, t'_k)$, $x(t) = x_k(t)$.
- For all $0 \le k \le N-1$, $x(t'_k) = (x_k(t'_k) + x_{k+1}(t_{k+1}))/2$.

The set of continuous traces of H is denoted $\mathcal{T}_c(H)$. If $I = [0, +\infty)$, τ_c is said to be infinite. The set of infinite continuous traces of H is denoted $\mathcal{T}_c^\infty(H)$.

Let us remark that at the continuous time point $\tau = t'_k = t_{k+1}$, which is the $(k+1)$th switching time, a hybrid trajectory takes on two (possibly distinct) continuous values:

---

[2] A function with k continuous derivatives is called a C^k function. Thus, a continuously differentiable function is a C^1 function.

the value $x_k(t'_k)$ before the switch, and the value $x_{k+1}(t_{k+1})$ after the switch. In the definition of a continuous trace, we arbitrarily choose to take the average of these two values. As we will see later, in the systems generated by the hybridization process, all the reset relations are given by the identity map restricted to the guard sets, and hence the above definition of $x(t'_k)$ implies the continuity of the function x because $x(t'_k) = x_k(t'_k) = x_{k+1}(t_{k+1})$.

We can now introduce the notion of *continuous reachable set*. Let $X_0 \subseteq \mathbb{R}^n$ be a set of continuous initial states. The continuous reachable set of H from X_0 within the time interval J is denoted by $\text{Reach}_c(H, X_0, J)$ and defined as follows. An element $y \in \mathbb{R}^n$ is in $\text{Reach}_c(H, X_0, J)$ if and only if there is a continuous trace $\tau_c = (I, x) \in \mathcal{T}_c(H)$ with $x(0) \in X_0$ such that there exists $t \in I \cap J : x(t) = y$. The reachable set for the time interval $[0, +\infty)$ is denoted by $\text{Reach}_c(H, X_0)$ for short.

2.2 Properties of hybrid systems

Hybrid systems have been mainly studied by researchers from computer science and control theory, who focus on different properties and approaches reflecting their background and view of hybrid systems. Control engineers are often interested in properties which are important in the design of a control system, such as stability, attraction, controllability, optimality... Computer scientists are interested in "sequential" properties expressed in some temporal logics, such as linear temporal logic (LTL) and computation tree logic (CTL) (see for example [24,43]). We will mention here some of the properties of hybrid systems that have been subject to investigation recently.

Stability and attraction. Roughly speaking, stability means that applying small perturbations on the system only results in small deviations in its responses. Depending on what is defined as perturbations and responses, this gives rise to a number of notions of stability. Stability of points is a widely used notion. Essentially, it requires that the state of the system remains in the neighborhood of some point under some small perturbations in the initial state. Given a hybrid system as in Definition 1, the (general) stability of a point $x^* \in \mathbb{R}^n$ can be defined as follows (see e.g. [51]):

$$\forall \varepsilon > 0, \ \exists \delta > 0, \ \forall \tau_c = (I, x) \in \mathcal{T}_c(H):$$
$$||x^* - x(0)|| < \delta \Rightarrow \ \forall t \in I : ||x^* - x(t)|| < \varepsilon.$$

where $|| \cdot ||$ denotes a norm on $\mathbb{R}^n$. Asymptotic stability of x^* additionally requires that

$$\exists \delta > 0, \ \forall \tau_c = (I, x) \in \mathcal{T}_c^\infty(H) : ||x^* - x(0)|| < \delta \Rightarrow \lim_{t \to +\infty} x(t) = x^*.$$

The geometric meaning of the stability of x^* is that for any neighborhood N_r of the point x^*, there exists a neighborhood N_p of the initial state such that no pertubation of the initial state in N_p makes the system's response leave N_r. In control theory, this stability of points for deterministic systems is often called *Lyapunov stability*. Stability is naturally one of the most important properties of control systems, since it guarantees that the system's evolution always stays close to some points representing the desired behaviors or the reference operation modes.

Besides stability, attraction is also of interest. For example, one can expect that the system not only stays close but also returns to the desired points (or trajectories), and this involves attraction properties. Intuitively, attraction of a point means that every neighborhood of the point must be reached after a finite amount of time. A set of such

points is called *attractor* and is formally defined as follows. An attractor A is a subset of the continuous state space $\mathbb{R}^n$ that satisfies the following two conditions [52]:

1. The set A is invariant:

$$\forall \tau_c = (I, x) \in \mathcal{T}_c^\infty(H) : x(0) \in A \Rightarrow \forall t \geq 0 : x(t) \in A.$$

2. The set A attracts a neighborhood B of itself (i.e. $A \subset B \subseteq \mathbb{R}^n$):

$$\forall \tau_c = (I, x) \in \mathcal{T}_c^\infty(H) : x(0) \in B \Rightarrow \lim_{t \to +\infty} d(x(t), A) = 0.$$

The first condition means that if the continuous state is in A, then it stays in A forever. In the second condition, the distance $d(y, A)$ from the point y to the set A is defined as $d(y, A) = \inf_{x \in A} ||y - x||$, and the set B is said to be in the *basin of attraction* of the attractor A. Two simple examples of attractors are the asymptotically stable fixed point and the asymptotically stable limit cycle. Note that stability and attraction are two closely related notions. Indeed, we can see that a point x^* is asymptotically stable if and only if it is stable and it is an attractor.

On the other hand, computer scientists approach the study of hybrid systems by applying the methodology of formal description and verification. They consider properties that can be formally described in some mathematical logics. As mentioned earlier, temporal logic is a popular formalism to describe properties involving the behavior of a system over time. In the following we briefly describe safety and eventuality properties, which are the most elementary classes of such properties.

Safety and eventuality.   Safety properties, and in particular invariance properties, have gained most of the attention in hybrid systems research. Intuitively, a safety property expresses that *nothing "bad" will happen*. Invariance properties are the simplest form of such properties, and an invariance property of the hybrid system H is of the form:

Starting from some set X_0 of initial states, all the continuous traces of H remain in a subset X_I of the continuous state space $\mathbb{R}^n$.

This is equivalent to a reachability property: *The reachable set* $\text{Reach}_c(H, X_0)$ *is included in* X_I. Therefore, invariance properties can be checked using reachable set computations. In temporal logic, the above invariance property is often expressed by the formula $\Box P$ where $\Box$ is the temporal quantifier *always*, and P is a predicate (formula) describing the set X_I. General safety properties are expressed by more complicated temporal logic formulae, but it is possible to express most interesting safety properties as invariance properties.

Eventuality properties assert that *something "good" must happen*. An example of such properties is:

Starting from some set X_0 of initial states, all the continuous traces of H eventually reaches a subset X_F of the continuous state space $\mathbb{R}^n$.

Note that a safety property is violated in finite time (since any infinite trace violating the property has a finite prefix that is "bad"). Hence, considering finite traces allow disproving safety properties. However, finite traces do not allow to disprove an eventuality property since there might be a finite trace that can be extended so that the resulting infinite trace satisfies this property.

2.3 Hybrid systems analysis: a brief review

Stability and attraction properties are mainly analyzed using various hybrid extensions of tools for continuous systems in control theory, such as Lyapunov functions [51], Poincaré maps [34]. It is well-known that the stability of a hybrid system is not guaranteed by the stability of all its continuous components, and one possible solution is to search for a global Lyapunov function in some fixed form, such as piecewise quadratic [21,35] or more general piecewise polynomial functions [47]. Stability of switched systems (which are systems consisting of a family of continuous dynamics and a rule to switch between them) has also been thoroughly studied in a number of publications (see [42] and references therein). The application of Poincaré maps to hybrid systems has also been explored, for example, in [59].

Formal analysis of hybrid systems is known to be a very difficult task due to the complexity and scale of the systems. Concerning specification, some temporal logics for hybrid systems have been developed (see for example [4,20]). Concerning verification, one is interested in designing an algorithm which, for a given hybrid system and a desired property, can answer after a finite number of steps whether the system satisfies the property. The decidability question is important in the algorithmic verification of hybrid systems, due to their infinite state space. A lot of research effort has been been devoted to the question of identifying the classes of hybrid systems and properties for which the verification problem is decidable. Temporal logic model-checking problems are not decidable for most general hybrid systems. However, decidability results have been proven for some particular classes of hybrid systems (see [4,54] for a survey). These classes must be restricted either in continuous dynamics (e.g. systems with piecewise constant derivatives [1,3,10,33]), or in discrete dynamics (e.g. [5,40]). Based on these results, various verification tools have been implemented, such as Kronos [60] and Uppaal [41] for timed automata, HyTech [32] for linear hybrid automata[3] and Requiem [40] for hybrid systems where linear differential equations have special eigenstructure and discrete dynamics can only have memoryless resets. The basis of these tools is a procedure for exactly characterizing and manipulating the set of all possible trajectories (using computer algebra tools for example).

However, this is no longer possible for more general hybrid systems. Indeed, even to prove simple safety properties, there exists no general exact reachable set computation method. For this reason, there has been growing interest in developing methods for the approximate representation and computation of sets of states and system traces, in particular with a focus on reachable set computation. These methods can be roughly classified into two categories. The methods of the first category try to approximate reachable sets as accurately as possible by tracking their evolution under the continuous flows using some set representation (such as polyhedra, ellipsoids, level sets). This results in a variety of approximation schemes (such as [6,13,15,17,19,27,28,36,39,44,57]), and implemented by a number of tools such as Coho[28], CheckMate [15], **d/dt** [7], VeriShift [13], HYSDEL [58], MPT [45], HJB toolbox [44]. Since accurate reachable set approximations are computationally expensive, the methods of the second category seek approximations that are sufficiently good to prove the property of interest (such as barrier certificates [46], polynomial invariants [56]). Abstraction methods for hybrid systems are also close in spirit to these

---

[3] In linear hybrid automata, continuous dynamics are described by linear constraints on the derivatives, such as $A\dot{x} \leq b$. They should not be confused with hybrid systems where continuous dynamics are described by linear differential questions.

methods. Indeed, their main idea is to start with a rough (and conservative) discrete approximation of a hybrid system and then iteratively refine it.[4] This refinement is often local in the sense that it uses the previous analysis results to determine where the approximation error is too large to prove the property (see for example [2,16,55]).

The literature on hybrid systems analysis is vast; for more details the reader is referred to recent proceedings of the conference HSCC (*Hybrid Systems: Computation and Control*). We finish this brief review by remarking that while many well-known properties of affine differential equations can be exploited to design relatively efficient algorithms, systems with nonlinear differential equations are much more difficult to analyze. For these systems, the existing tools can only handle a small number of continuous variables. This motivated us to search for a method to deal with nonlinear systems using tools available for simpler systems.

3 Principles of hybridization

Intuitively, the main idea of hybridization is to approximate the complex continuous dynamics of a system by a collection of simpler continuous dynamics. Here, by "simpler" we mean the types of dynamics that can be analyzed more easily and efficiently. For example, we can approximate a nonlinear continuous dynamics by a piecewise affine dynamics. The collection of simpler continuous dynamics indeed defines a hybrid system. Then, the analysis of the resulting hybrid system can provide knowledge about the behavior of the original system. Hence, one can see another utility of hybrid systems: they can be used not only as a mathematical model but also as an approximation method.

For simplicity of presentation, we explain the principle of the hybridization approach for continuous, autonomous (i.e. without inputs) dynamical systems. However, the approach can be extended to handle continuous dynamical systems with inputs as well as hybrid systems, which will be discussed later in this section. We consider a nonlinear continuous dynamical system D that is defined on a domain $\Omega \subseteq \mathbb{R}^n$ by a differential equation of the form:

$$\dot{x}(t) = f(x(t)), \quad x(t) \in \Omega, \quad t \geq 0.$$

This continuous dynamical system can be seen as a hybrid system with a unique location (denoted by ω) and no discrete transition:

$$D = (\{\omega\}, n, 0, \emptyset, \{(\emptyset, f)\}, \{\Omega\}, \emptyset, \emptyset).$$

As mentioned earlier, the idea of hybridization consists in approximating the nonlinear vector field f by a hybrid system with a collection of simpler (e.g. constant or affine) vector fields. To do so, the domain Ω of the dynamical system D is split into several regions that form a *mesh* of Ω. Then, with each element of this mesh, we associate a simple approximate vector field. This defines a global approximate vector field all over the domain Ω, that we call *composite vector field* . In particular, in order for this approximation to be useful in terms of preserving properties of interest, we want it to be a *conservative approximation* of the original vector field f. The reasons for this will become clear in the next section. This composite vector field corresponds

---

[4] Most proofs of decidability result for certain classes of hybrid systems are often based on the existence of a finite discrete abstraction (see [4]).

to a hybrid system, which we call[5] a *hybridization* of D. Let us now formalize these notions.

Definition 5 (Mesh) A *mesh* of the domain Ω is a collection $\mathcal{M} = \{\mathcal{M}_l \mid l \in L\}$ of closed subsets of Ω such that

- $\bigcup_{l \in L} \mathcal{M}_l = \Omega$.
- For all $l \neq l' \in L$, $\mathcal{M}_l \cap \mathcal{M}_{l'} = \partial \mathcal{M}_l \cap \partial \mathcal{M}_{l'}$, where $\partial \mathcal{M}_l$ denotes the boundary of the set $\mathcal{M}_l$.

The *size* of the mesh is defined as:

$$\sigma(\mathcal{M}) = \sup_{l \in L} \sup_{x, x' \in \mathcal{M}_l} ||x - x'||$$

where $|| \cdot ||$ denotes the infinity norm and is defined as follows: for a n-dimensional vector y, $||y|| = \max_{i \in \{1,2,\ldots,n\}} |y_i|$.

The first condition guarantees that the domain Ω is covered by the mesh. The second condition indicates that the elements of the mesh have disjoint interiors. It should be noted that in the above definition we use the infinity norm, but the reasoning throughout the paper can work with any norm.

Definition 6 (Composite vector field) A *composite vector field* on the domain Ω is a collection $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) \mid l \in L\}$ where $\mathcal{M} = \{\mathcal{M}_l \mid l \in L\}$ is a mesh of Ω and for all $l \in L$, $U_l \subseteq \mathbb{R}^n$ and $f_l : \mathbb{R}^n \to \mathbb{R}^n$.

A composite vector field $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) \mid l \in L\}$ on the domain Ω implicitly defines a hybrid system $H(\mathcal{F}) = (L, n, n, E, F, \mathcal{M}, G, R)$ where

- the set of discrete locations is L.
- the dimension of the continuous state space is n.
- the dimension of the continuous input is n.
- the set of discrete transitions is given by

$$(l, l') \in E \iff \partial \mathcal{M}_l \cap \partial \mathcal{M}_{l'} \neq \emptyset.$$

- the collection of vector fields $F = \{F_l \mid l \in L\}$ where $F_l = (U_l, f'_l)$ with f'_l defined as: $f'_l(x, u) = f_l(x) + u$.
- the invariants are the domains of the mesh elements $\mathcal{M}_l$.
- the guards are given by

$$\forall e = (l, l') \in E, \quad G_e = \partial \mathcal{M}_l \cap \partial \mathcal{M}_{l'}.$$

- the resets relations are the identity map restricted to the guard sets:

$$\forall e \in E, \quad R_e = \{(x, x) \mid x \in G_e\}.$$

Such a composite vector field is used to approximate the vector field f of an autonomous system, and the role of the input is to model the error between values of f in each mesh cell $\mathcal{M}_l$ and its approximation f_l in that cell.

---

[5] In this paper we use the term hybridization to mean the approximate system and sometimes the approximation process as well.

Definition 7 A composite vector field $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) | \ l \in L\}$ on the domain Ω is a *conservative approximation* of the vector field f if

$$\forall l \in L, \quad x \in \mathcal{M}_l, \quad \exists u \in U_l, \quad f(x) = f_l(x) + u.$$

If $\mathcal{F}$ is a conservative approximation of the vector field f then the hybrid system $H(\mathcal{F})$ is said to be a *hybridization* of the system D. The *precision of the hybridization* is given by:

$$\pi(\mathcal{F}, f) = \sup_{l \in L} \left(\sup_{x \in \mathcal{M}_l, \, u \in U_l} \|f_l(x) + u - f(x)\| \right).$$

Extension to continuous systems with input and to hybrid systems. It is straightforward to extend the above definitions to continuous systems with input by choosing appropriately the sets U_l of the hybridization, which allows taking into account the effect of the input of the system. As an example, we consider a continuous system with input of the form $\dot{x}(t) = f(x(t)) + v(t)$ where $v(t) \in V$. Let $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) | \ l \in L\}$ be a hybridization of the corresponding autonomous system $\dot{x}(t) = f(x(t))$. Then, the hybridization of the system with input can be defined as $\overline{\mathcal{F}} = \{(\mathcal{M}_l, \overline{U}_l, f_l) | \ l \in L\}$ where the input set $\overline{U}_l = U_l \oplus V$, and $\oplus$ denotes the Minkowski sum.

Regarding hybrid systems, the global approximate system can be obtained by a composition of the hybridizations of the continuous dynamics in each location. The hybridization process can also be partial in the sense that it is used only in some locations.

4 Approximation properties of hybridizations

In order for a hybridization to be a useful approximation, a question that arises is whether the hybridization preserves the properties of interest. In the following we show several important approximation properties of the hybridization defined in the previous section. We also discuss how these results can be used in the analysis of complex continuous and hybrid systems. Let us make the following assumption:

Assumption 1 (Finite Variability [12])*For all continuous trace $(I, x) \in \mathcal{T}_c(D)$, for all interval of finite length $J \subseteq I$, the function x can move from one cell of the mesh $\mathcal{M}$ to another only a finite number of times during time interval J.*

As we will see, this assumption is necessary so that the corresponding continuous traces of the hybridization do not exhibit a Zeno behaviors. This assumption is necessary for our approach but also for all discrete abstraction processes based on a partition of the state space [2,4,16,55]. Let us remark that in practice, the finite variability assumption generally holds. An interesting investigation is to determine the conditions that the vector field f must satisfy so that the finite variability assumption holds. This is not done here as it would require complex mathematical considerations and it is preferable to focus our attention on the main scope of the paper.

4.1 Trace inclusion and approximation

The first approximation property involves the conservativeness of the hybridization in terms of sets of traces.

Theorem 1 *Let $H(\mathcal{F})$ be a hybridization of the dynamical system D. Then, the set of continuous traces of D is included in the set of continuous traces of $H(\mathcal{F})$, that is*

$$\mathcal{T}_c(D) \subseteq \mathcal{T}_c(H(\mathcal{F})).$$

Proof Let $(I, x) \in \mathcal{T}_c(D)$. We assume that I is of the form $[0, T]$, and the situation where $I = [0, +\infty)$ can be handled in a similar way. Since $\bigcup_{l \in L} \mathcal{M}_l = \Omega$, there exists a piecewise constant function $q : I \to L$ such that

$$\forall t \in I, \quad x(t) \in \mathcal{M}_{q(t)}.$$

Let $t_0 = 0$ and let $\{t_k | 1 \leq k \leq N\}$ be the sequence of instants at which the value of the function q changes. From the finite variability assumption, we have that necessarily N is finite. For $0 \leq k \leq N$, q_k denotes the constant value of $q(t)$ on the open interval (t_k, t_{k+1}). Given an interval $I_k = [t_k, t_{k+1}]$, the function $x_k : I_k \to \mathbb{R}^n$ is such that for all $t \in I_k$, $x_k(t) = x(t)$. We now show that $(\{I_k | 0 \leq k \leq N\}, \{q_k | 0 \leq k \leq N\}, \{x_k | 0 \leq k \leq N\})$ is a hybrid trace of $H(\mathcal{F})$.

For $0 \leq k \leq N$ and for all $t \in (t_k, t_{k+1})$, $x(t) \in \mathcal{M}_{q_k}$. By continuity of x and closedness of $\mathcal{M}_{q_k}$, it then follows that: for all $t \in I_k = [t_k, t_{k+1}]$, $x(t) \in \mathcal{M}_{q_k}$. We define an input function $u_k : I_k \to \mathbb{R}^n$ as:

$$\forall t \in I_k, \quad u_k(t) = f(x_k(t)) - f_{q_k}(x_k(t)).$$

Note that since the composite vector field $\mathcal{F}$ is a conservative approximation of f, $u_k(t) \in U_{q_k}$ for all $t \in I_k$. Moreover,

$$\forall t \in I_k, \quad \dot{x}_k(t) = f(x_k(t)) = f_{q_k}(x_k(t)) + u_k(t).$$

Furthermore, $x_k(t_{k+1}) \in \mathcal{M}_{q_k} \cap \mathcal{M}_{q_{k+1}}$, thus $e_k = (q_k, q_{k+1}) \in E$ and $x_k(t_{k+1}) \in G_{e_k}$. In addition, since $x_k(t_{k+1}) = x(t_{k+1}) = x_{k+1}(t_{k+1})$, we have that $(x_k(t_{k+1}), x_{k+1}(t_{k+1})) \in R_{e_k}$. Hence, $(\{I_k | 0 \leq k \leq N\}, \{q_k | 0 \leq k \leq N\}, \{x_k | 0 \leq k \leq N\})$ is in the set $\mathcal{T}_h(H(\mathcal{F}))$ of hybrid traces of $H(\mathcal{F})$. We end the proof by remarking that (I, x) is the continuous trace of $H(\mathcal{F})$ associated with the hybrid trace $(\{I_k | 0 \leq k \leq N\}, \{q_k | 0 \leq k \leq N\}, \{x_k | 0 \leq k \leq N\})$. □

Let us remark that the hybridization $H(\mathcal{F})$ is indeed a simulation of the original system D. An important consequence of this theorem is that the hybridization $H(\mathcal{F})$ can be used to check safety and eventuality properties of the continuous dynamical system D [9,25,33]. Indeed, if these properties hold for the set of *all* continuous traces of the hybridization $H(\mathcal{F})$, then it follows from the inclusion relation that they hold for the set of *all* continuous traces of the dynamical system D. In terms of temporal logics, this means that the universal fragment $\forall$ CTL* [24] is preserved by the hybridization. Since LTL properties are part of $\forall$ CTL* [24], they are also preserved by the hybridization.

We have seen that the hybridization method allows over-approximating the set of continuous traces of D. Now to measure the quality of over-approximation, we use the distance between the set of continuous traces of $H(\mathcal{F})$ and that of D.

Theorem 2 *We assume that f is λ-Lipschitz on Ω, that is*

$$\forall x, z \in \Omega, \quad \|f(x) - f(z)\| \leq \lambda \|x - z\|.$$

Then, for all $(I, x) \in \mathcal{T}_c(D)$, $(J, z) \in \mathcal{T}_c(H(\mathcal{F}))$, such that $x(0) = z(0)$, the following inequality holds:

$$\forall t \in I \cap J, \quad \|x(t) - z(t)\| \leq \frac{\pi(\mathcal{F}, f)}{\lambda}(e^{\lambda t} - 1). \tag{1}$$

Proof The proof relies heavily on the fundamental inequality, and for clarity we first recall this theorem.

Theorem 3 *Let Ω be a subset of $\mathbb{R}^n$, let $f : \Omega \to \mathbb{R}^n$ be a λ-Lipschitz vector field. Let $x : I \to \mathbb{R}^n$ and $z : J \to \mathbb{R}^n$ be piecewise differentiable functions such that $I \cap J$ is not empty and contains 0 and for all $t \in I \cap J$, $x(t) \in \Omega$ and $z(t) \in \Omega$. If in addition, for all $t \in I \cap J$, where x is differentiable $\|\dot{x}(t) - f(x(t))\| \leq \alpha$, and for all $t \in I \cap J$, where z is differentiable $\|\dot{z}(t) - f(z(t))\| \leq \beta$, then the following inequality holds:*

$$\forall t \in I \cap J, \quad \|x(t) - z(t)\| \leq \|x(0) - z(0)\|e^{\lambda|t|} + \frac{\alpha + \beta}{\lambda}(e^{\lambda|t|} - 1). \tag{2}$$

The fundamental inequality is a central theorem of the theory of differential equations. A detailed proof can be found in [23]. We now proceed with the proof of Theorem 2. Let $(I, x) \in \mathcal{T}_c(D)$, then x is differentiable on I and for all $t \in I$ $\|\dot{x}(t) - f(x(t))\| = 0$. Let $(J, z) \in \mathcal{T}_c(H(\mathcal{F}))$. Let $(\{J_k| \ 0 \leq k \leq N\}, \{q_k| \ 0 \leq k \leq N\}, \{z_k| \ 0 \leq k \leq N\})$ be the hybrid trace of $H(\mathcal{F})$ associated with (J, z). For all $0 \leq k \leq N$, for all piecewise continuous input $u_k : J_k \to U_{q_k}$ such that for all $t \in J_k$ where u_k is continuous,

$$\|\dot{z}(t) - f(z(t))\| = \|f_{q_k}(z(t)) + u_k(t) - f(z(t))\| \leq \pi(\mathcal{F}, f).$$

Then, the inequality (1) follows from the application of the fundamental inequality.
$\square$

The above theorem states that the distance between the set of continuous traces of $H(\mathcal{F})$ and that of D depends linearly on the precision $\pi(\mathcal{F}, f)$. The theorem thus shows that one can approximate the set of traces of the system D with an arbitrarily small error by using an appropriate hybridization (i.e. with a sufficiently good precision). This result was also used in the previous works [22,26] to justify the application of the hybridization method to approximate solutions of differential equations. Indeed, assuming that each input sets U_l contains the origin, that is $0 \in U_l$, one can effectively compute an approximate solution of the equation $\dot{x} = f(x)$ by computing the continuous traces of $H(\mathcal{F})$ corresponding to the input functions $\{u_k| \ 0 \leq k \leq N\}$ satisfying $u_k(t) = 0$ for all $t \in J_k$.

4.2 Preservation of attractors

Theorems 1 and 2 can also be used to prove another important property of hybridizations involving the preservation of attractors. Let us assume that the dynamical system D has an attractor A, attracting a compact set B. In addition, let us assume that the following conditions hold:

1. $\forall x_0 \in B, \exists (I, x) \in \mathcal{T}_c^\infty(D) : x(0) = x_0$.
2. $\exists \delta > 0 : \mathcal{N}(A, \delta) = \{x \in \mathbb{R}^n| \ \exists x_a \in A, \ \|x - x_a\| \leq \delta\} \subseteq B$.

The set $\mathcal{N}(A, \delta)$ is called δ-neighborhood of A.

Theorem 4 *For all $\varepsilon \in (0, \delta]$, there exists $\nu > 0$, such that if $\pi(\mathcal{F}, f) \leq \nu$, then there exists a set $A(\mathcal{F})$, which is an attractor for the hybrid system $H(\mathcal{F})$ and such that*

$$A \subseteq A(\mathcal{F}) \subseteq \mathcal{N}(A, \varepsilon).$$

Moreover, B is in the basin of attraction of $A(\mathcal{F})$.

The theorem states that attractors are preserved by hybridization, and one can thus use the hybridization method to check attraction properties of continuous and hybrid systems.

Proof Since A attracts the compact set B, there exists a time $T > 0$ such that

$$\forall (I, x) \in \mathcal{T}_c^\infty(D), \quad x(0) \in B \implies \forall t \geq T, \quad x(t) \in \mathcal{N}(A, \varepsilon/2). \tag{3}$$

We define

$$v = \frac{\varepsilon \lambda}{2(e^{\lambda 2T} - 1)}.$$

Let us assume that $\pi(\mathcal{F}, f) \leq v$. Let $(J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$ such that $z(0) \in B$. Then, there exists $(I, x) \in \mathcal{T}_c^\infty(D)$ such that $x(0) = z(0)$. From Theorem 2, we have that

$$\forall t \in [0, 2T], \quad \|x(t) - z(t)\| \leq \frac{\pi(\mathcal{F}, f)}{\lambda}(e^{\lambda 2T} - 1) \leq \varepsilon/2. \tag{4}$$

Then, from the equation (3), it follows that

$$\forall (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), \quad z(0) \in B \implies \forall t \in [T, 2T], \quad z(t) \in \mathcal{N}(A, \varepsilon). \tag{5}$$

Let us now show by induction that this actually holds for all $t \geq T$. We first assume that the previous equation holds for all $t \in [T, kT]$ for some $k \geq 2$. Let $(J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$ such that $z(0) \in B$, let $\bar{t} \in [kT, (k+1)T]$, let $(\bar{I}, \bar{z})$ be defined by

$$\bar{I} = [0, +\infty), \quad \bar{z}(t) = z(\bar{t} - T + t).$$

Intuitively, $\bar{z}$ is the suffix of z obtained by truncating off the initial length $\bar{t} - T$, and starting with $\bar{z}(0) = z(\bar{t} - T)$. It is easy to show that $(\bar{I}, \bar{z}) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$. Moreover, $\bar{z}(0) = z(\bar{t} - T)$. Since $\bar{t} - T \in [T, kT]$, we obtain: $\bar{z}(0) \in \mathcal{N}(A, \varepsilon) \subseteq B$. From the equation (5), it follows that $\bar{z}(T) \in \mathcal{N}(A, \varepsilon)$. Since $\bar{z}(T) = z(\bar{t})$, we proved by induction that

$$\forall (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), \quad z(0) \in B \implies \forall t \in [T, +\infty), \quad z(t) \in \mathcal{N}(A, \varepsilon). \tag{6}$$

Now let us define the set $A(\mathcal{F})$ by

$$A(\mathcal{F}) = \{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in B, t \geq T\}$$
$$\cup \{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in A, t \geq 0\}$$
$$= \{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in B, t \geq T\}$$
$$\cup \{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in A, t \in [0, T]\} \tag{7}$$

Clearly $A \subseteq A(\mathcal{F})$. Moreover, from the Eq. (6)

$$\{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in B, t \geq T\} \subseteq \mathcal{N}(A, \varepsilon). \tag{8}$$

Let $(J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$, such that $z(0) \in A$, there exists $(I, x) \in \mathcal{T}_c^\infty(D)$ such that $x(0) = z(0)$. Let $t \in [0, T]$; since A is an attractor for D, $x(t) \in A$. Then, it follows from equation (4) that $z(t) \in N(A, \varepsilon/2)$. Hence,

$$\{z(t) | (J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F})), z(0) \in A, t \in [0, T]\} \subseteq \mathcal{N}(A, \varepsilon/2). \tag{9}$$

Thus, we have proved that $A(\mathcal{F}) \subseteq N(A, \varepsilon)$. Next, we prove that $A(\mathcal{F})$ is an attractor for $H(\mathcal{F})$ and that B is in its basin of attraction. Let $(J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$, such that $z(0) \in A(\mathcal{F})$. There are two possible situations. Either there exists $(\bar{J}, \bar{z}) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$

and $\bar{t} \geq T$ such that $\bar{z}(0) \in B$ and $\bar{z}(\bar{t}) = z(0)$, or there exists $(\bar{J}, \bar{z}) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$ and $\bar{t} \geq 0$ such that $\bar{z}(0) \in A$ and $\bar{z}(\bar{t}) = z(0)$. We consider only the first situation, and the second one can be handled in a similar way. We define $(\tilde{J}, \tilde{z})$ as

$$\tilde{J} = [0, +\infty), \quad \tilde{z}(t) = \begin{cases} \bar{z}(t), & \text{if } t \in [0, \bar{t}] \\ z(t - \bar{t}), & \text{if } t \in [\bar{t}, +\infty) \end{cases} \tag{10}$$

We can show that $(\tilde{J}, \tilde{z}) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$, thus from the definition of $A(\mathcal{F})$, for all $\tilde{t} \geq T$, $\tilde{z}(\tilde{t}) \in A(\mathcal{F})$. Therefore, since $\bar{t} \geq T$, for all $t \geq 0$, $z(t) = \tilde{z}(t + \bar{t}) \in A(\mathcal{F})$. This implies that $A(\mathcal{F})$ is invariant for $H(\mathcal{F})$. It now remains to show that it attracts B. Let $(J, z) \in \mathcal{T}_c^\infty(H(\mathcal{F}))$, such that $z(0) \in B$. From the definition of $A(\mathcal{F})$, for all $t \geq T$, $z(t) \in A(\mathcal{F})$. It then follows that $\lim_{t \to +\infty} d(z(t), A(\mathcal{F})) = 0$. □

Relation to previous work. Before continuing to explain how to actually construct a hybridization, let us discuss the relation of the method to some previous work. Regarding the approximation and abstraction categories discussed in Sect. 2.3, our hybridization method has elements of both. On one hand, it provides approximations with error bounds. It should be noted that for safety verification it is sufficient to prove the property on an over-approximation of the original system; however for other problems, such as controller synthesis where under-approximations are used, the accuracy criterion is important since we do not want to disregard too much controllable behavior. On the other hand, our method has some similarity with the abstraction methods mentioned in Sect. 2.3, since the approximate systems we construct are indeed abstractions of the original system, which are more precise at the price of being more complex. Therefore, the refinement ideas [2,16,55] can be straightforwardly applied in our approach.

Our hybridization method also has some common elements with the viability algorithms [50] for approximating viability kernels of differential inclusions. The approximation in the viability algorithms involves not only a space discretization but also a time discretization, which results in a hybrid system in discrete time. In our hybridization, the resulting approximate system is in continuous time. Similar hybridization ideas were previously explored in [25,33,48,53] where the approximate systems are systems with piecewise constant slopes or rectangular inclusions. As mentioned earlier, similar ideas have been used for numerical integration of nonlinear differential equations [22,26]. On the other hand, the reachability method proposed in [28] uses linear approximation in each integration step to obtain better approximations of the reachable sets in 2 dimensions. In [29], a control problem for a class of piecewise linear systems, similar to our affine hybridizations, is solved in terms of reachability conditions. In [49], a hybridization approach is used to solve optimal control problems.

5 Effective construction of hybridizations

In the previous section, we have shown that hybridizations can be used to analyze a fairly large class of properties. In this section, we deal with the question of constructing hybridizations for systems with possibly complex nonlinear continuous dynamics. We propose two methods to do so, and a bound on the precision of the hybridization is also provided. Let us recall the continuous system under study:

$$\dot{x}(t) = f(x(t)), \quad x(t) \in \Omega,\ t \geq 0.$$

5.1 Affine hybridization

The first hybridization construction method, called *affine hybridization*, computes a piecewise affine vector field by partitioning the state space into simplices. The use of such approximate vector fields is motivated by a large choice of available methods for reachability computation of piecewise affine systems (see for instance [6,15,27]). It is worth emphasizing that, besides reachability properties, the hybridization can be used to verify other classes of properties of the original system, provided that we are equipped with a tool to verify these properties on the hybridization.

5.1.1 Constructing the mesh

We start by constructing a simplicial mesh $\mathcal{M} = \{\mathcal{M}_l | \; l \in L\}$ over the set Ω where each mesh element $\mathcal{M}_l$ is a simplex. We recall that a simplex in $\mathbb{R}^n$ is the convex hull of $(n+1)$ affinely independent points in $\mathbb{R}^n$. A simplicial mesh can be easily obtained from a rectangular mesh $\{\mathcal{H}_j | \; j \in J\}$ over Ω, generated by a uniform grid of size h. Each mesh element $\mathcal{H}_j$, which is a hypercube with edge length h, is then partitioned into simplices as follows. The explanation here holds for the hypercube $[0,1]^n$ but the procedure generalizes easily to any hypercube. Let Θ denote the set of permutations of $\{1, \ldots, n\}$. For all $\theta = (i_1, \ldots, i_n) \in \Theta$, the set

$$\mathcal{S}_\theta = \{x \in [0,1]^n : \; 0 \le x_{i_1} \le \cdots \le x_{i_n} \le 1\}$$

is a simplex of $\mathbb{R}^n$. Moreover, $\{\mathcal{S}_\theta | \; \theta \in \Theta\}$ indeed defines a simplicial mesh of $[0,1]^n$. A proof of this can be found in [38].

It is easy to see that the resulting simplicial mesh has the size $\sigma(\mathcal{M}) = h$. We note that the cardinality of Θ is the factorial $n!$.

This mesh generation method is very simple and, moreover, memory efficient since it can be done in the course of the analysis of the hybridization. For example, when performing reachability analysis by advancing the set of solutions numerically step by step, we only need to create new mesh elements around the solutions. Therefore, we do not need to generate the whole mesh before we start any computation (unlike using a Delaunay triangulation whose dynamic construction is expensive).

5.1.2 Defining the hybridization

Once we have our simplicial mesh $\mathcal{M}$, we define a composite vector field $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) \mid l \in L\}$ as follows. For each mesh element $\mathcal{M}_l$, we denote by $V(\mathcal{M}_l)$ the set of its vertices. We define f_l as an affine map of the form: $f_l(x) = A_l x + b_l$ (A_l is a matrix of size $n \times n$ and $b_l \in \mathbb{R}^n$) such that f_l interpolates the function f at the vertices of $\mathcal{M}_l$. More precisely, $f(v) = f_l(v)$ for all $v \in V(\mathcal{M}_l)$. An important advantage of this hybridization method is that using simplicial meshes the affine interpolant f_l is uniquely determined, since each simplicial mesh element has $(n+1)$ vertices.

Let us now define U_l so that $\mathcal{F}$ is a conservative approximation of the original vector field f, as in Definition 7. To this end, we define the interpolation error as:

$$\varepsilon(h) = \sup_{l \in L} \; \sup_{x \in \mathcal{M}_l} \|f(x) - f_l(x)\|.$$

We write this error as a function of h to emphasize that it depends on the mesh size. Suppose that we can find an upper bound of $\varepsilon(h)$, denoted by $\bar{\varepsilon}(h)$. Then, we can

choose the input value set U_l to be the ball (i.e. a hypercube for the infinity norm) that is centered at the origin and has radius $\bar{\varepsilon}(h)$. We estimate this bound for two cases: the vector field f is Lipschitz and f is a C^2 function. The proofs can be found in Appendix.

Proposition 1 *If f is λ-Lipschitz, then*

$$\varepsilon(h) \leq h \frac{2n\lambda}{n+1} = \bar{\varepsilon}(h).$$

An important remark is that the second partial derivatives of the affine approximation f_l vanish; therefore, if f is a C^2 function, we can obtain a better error bound. Exceptionally, to write the second partial derivatives of f, we use superscripts to indicate the components of f, that is $f = (f^1, f^2, \ldots, f^n)$.

Proposition 2 *If f is C^2 on Ω with bounded second order derivatives then*

$$\varepsilon(h) \leq \frac{Kn^2}{2(n+1)^2}h^2 = \bar{\varepsilon}(h)$$

where

$$K = \max_{i \in \{1,\ldots,n\}} \sup_{x \in \Omega} \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \left| \frac{\partial^2 f^i(x)}{\partial x_{p_1} \partial x_{p_2}} \right|.$$

The main drawback of affine hybridization is the cardinality of the simplicial partition of Ω. We have seen that $n!$ simplices are needed to partition a hypercube in $\mathbb{R}^n$. In the next section, we propose a similar method but based on a rectangular mesh. As a consequence, the complexity of the resulting hybridization is lower and therefore systems in higher dimensions can be considered.

5.2 Multi-affine hybridization

Rectangular meshes and multi-affine interpolation can be thought of as a simple generalization of affine interpolation. The technique is most easily explained in two dimensions, but the case of higher dimensions is analogous in every way.

5.2.1 Constructing the mesh

We assume that the state space is a rectangle $\Omega = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \subset \mathbb{R}^2$, let

$$\underline{x}_1 = x_1^1 < x_1^2 < \cdots < x_1^m = \bar{x}_1, \quad \underline{x}_2 = x_2^1 < x_2^2 < \cdots < x_2^p = \bar{x}_2$$

such that for all $1 \leq i \leq m$, $x_1^{i+1} - x_1^i = h$ and for all $1 \leq j \leq m$, $x_2^{j+1} - x_2^j = h$. This defines a uniform mesh with $(m-1)(p-1)$ elementary rectangles and mp interpolating points (x_1^i, x_2^j). Hence, the index set L of the mesh $\mathcal{M}$ is $L = \{1, 2, \ldots, (m-1)(p-1)\}$.

5.2.2 Defining the hybridization

We now show how to define the piecewise multi-affine vector field $\mathcal{F} = \{(\mathcal{M}_l, U_l, f_l) \mid l \in L\}$. For each $l \in L$, we define a multi-linear function that

interpolates f at the vertices of the corresponding rectangle. Suppose that the rectangle is $[x_1^i, x_1^{i+1}] \times [x_2^j, x_2^{j+1}]$,

$$
\begin{aligned}
f_l(x) = {} & \frac{(x_2^{j+1} - x_2)}{(x_2^{j+1} - x_2^j)} \frac{(x_1^{i+1} - x_1)}{(x_1^{i+1} - x_1^i)} f(x_1^i, x_2^j) \\
& + \frac{(x_2^{j+1} - x_2)}{(x_2^{j+1} - x_2^j)} \frac{(x_1 - x_1^i)}{(x_1^{i+1} - x_1^i)} f(x_1^{i+1}, x_2^j) \\
& + \frac{(x_2 - x_2^j)}{(x_2^{j+1} - x_2^j)} \frac{(x_1^{i+1} - x_1)}{(x_1^{i+1} - x_1^i)} f(x_1^i, x_2^{j+1}) \\
& + \frac{(x_2 - x_2^j)}{(x_2^{j+1} - x_2^j)} \frac{(x_1 - x_1^i)}{(x_1^{i+1} - x_1^i)} f(x_1^{i+1}, x_2^{j+1})
\end{aligned}
$$

To see how this definition can be generalized to higher dimensions, we remark that the above function f_l can be thought of as the result of iteratively applying an affine interpolation on each dimension. More concretely, we denote by $\lambda_f(k; x_k^i, x_k^{i+1})$ the function obtained by applying an affine interpolation of f on the kth dimension at two points x_k^i and x_k^{i+1}. Let

$$
p(x) = \lambda_f(1; x_1^i, x_1^{i+1}) \quad \text{and} \quad q(x) = \lambda_f(2; x_2^j, x_2^{j+1}).
$$

Then, the function f_l for the rectangle $[x_1^i, x_1^{i+1}] \times [x_2^j, x_2^{j+1}]$ can be rewritten as:

$$
f_l(x) = \lambda_p(2; x_2^j, x_2^{j+1}) = \lambda_q(1; x_1^i, x_1^{i+1}).
$$

As in the case of affine interpolation over a simplicial mesh, the piecewise multi-affine function f_l which interpolates f at all mesh vertices is unique and continuous over Ω. The following proposition gives a bound on the multi-affine interpolation error, which is also quadratic in the mesh size if f is C^2. The proof can be found in [30].

Proposition 3 *If f is C^2 on Ω with bounded second order derivatives then*

$$
\varepsilon(h) = \sup_{x \in \Omega} \|f(x) - f_l(x)\| \leq \frac{M}{8} h^2 = \bar{\varepsilon}(h)
$$

where $M = \sum_{i=1}^{i=n} \|D_{x_i} f\|^2$ with $D_{x_i} f = \frac{\partial}{\partial x_i} J$ and J is the Jacobian matrix of f.

Again, the above error bound $\bar{\varepsilon}(h)$ is used to define the input value set U_l of the composite vector field $\mathcal{F}$.

Besides the advantage of rectangular meshes over simplicial meshes with respect to the complexity of the hybridization, the computation of set intersection occurring at the transitions between adjacent mesh elements is also simpler (due to the axis-aligned boundary between the mesh elements). Then, reachability computation for the approximate piecewise multi-affine system can be done using the method proposed in [8]. The main idea of this method is to project away some continuous variables and consider them as inputs taking values in the corresponding mesh element ranges. This results in a piecewise bilinear system with uncertain inputs, which is then handled by a reachability analysis method using the Maximum principle. Alternatively, one can apply our recent method for polynomial systems [17] which exploits the geometric properties of polynomial maps without resorting to variable projection.

5.3 Convergence

Thus, we have completely defined a piecewise affine or multi-affine vector field $\mathcal{F}$. The following result is a direct consequence of the above construction of the hybridization.

Theorem 5 *The piecewise affine vector field $\mathcal{F}$ is a conservative approximation of f, and the precision $\pi(\mathcal{F}, f)$ is bounded by $2\overline{\varepsilon}(h)$.*

We can see that the composite vector field $\mathcal{F}$ converges uniformly to f with respect to the space discretization size, that is the mesh size h. Therefore, regarding reachability analysis, Theorem 2 shows that the continuous traces of $H(\mathcal{F})$ converges to the continuous traces of the original system with the same rate as the composite vector field $\mathcal{F}$ converges to f. In the case where f is a C^2 function, this rate is quadratic. As mentioned earlier, reachability analysis of $H(\mathcal{F})$ can be done using any of the available methods for piecewise affine or multi-affine systems.

However, it should be noted from (2) that the error bound diverges exponentially in the duration of the traces. Hence, when the verification requires the computation of the reachable set for a long interval of time, the error bound given by Theorem 2 might be large. However, for a system with an attractor that attracts all the traces of the system (that is, the basin of attraction is the whole state space), using Theorem 4, we can prove the following convergence result concerning the reachable set approximation. Given a mesh size h, we denote by $\mathcal{F}_h$ the composite vector field constructed as shown in Sect. 5.1.2.

Corollary 1 *Let $X_0 \subset \Omega$ be a compact set of initial states. If the dynamical system D has an attractor which attracts all the trajectories of D, then there exists a mesh size h' such that for all $h \leq h'$ the distance between the reachable set of D and that of the hybridization $H(\mathcal{F}_h)$ is bounded. Moreover,*

$$\lim_{h \to 0} d_H \left(\text{Reach}_c(D, X_0), \text{Reach}_c(H(\mathcal{F}_h), X_0) \right) = 0$$

where d_H denotes the Hausdorff distance.

6 Applications

We have implemented the hybridization method and integrated it in the verification tool **d/dt** [6,7]. The tool **d/dt** was initially designed to handle the following class of hybrid systems: the continuous dynamics are linear and possibly with bounded uncertain inputs, and the guard and staying conditions associated with discrete transitions and locations are described by linear contraints (or convex polytopes). The main functionality of the tool is the verification of safety and reachability properties, based on reachable set computations. The integration of the hybridization method in the tool extended its scope to hybrid systems with nonlinear continuous dynamics. We have experimented this new functionality of the tool on various applications [8,9,18]. In this section, we illustrate the method with some examples.

Before continuing, we remark that in the implementation of the hybridization method, we use a dynamical construction of the mesh, that is, only the cells around the trajectories under exploration are created. This is an important advantage of the hybridization approach, compared to the grid-based approaches that requires the

whole mesh to be constructed before starting the computation (such as [44,50]). However, in our approach, treating the transitions between the cells requires computing the intersection with the cell boundaries, which is expensive especially in high dimensions. We are currently working on a method that reduces this intersection computation by defining a transient dynamics in a neighborhood of the cell boundaries.

The Van der Pol System

The first example is the two dimensional Van der Pol oscillator:

$$\begin{cases} \dot{x}(t) = y(t) \\ \dot{y}(t) = y(t)(1 - x^2(t)) - x(t) \end{cases}$$

It is well-known that the Van der Pol oscillator has a limit cycle which attracts all the trajectories of the system. Thus, it satisfies the condition of Corollary 1. We used a triangular mesh of size $h = 0.05$ to construct the hybridization. In this example, we use the bound in Proposition 2. An input was added to account for the approximation error. The reachable set computed by the hybridization method is shown in Fig. 1, where the dotted set represents the set of initial values. The figure shows that the reachable set indeed contains the limit cycle.

Biquad lowpass filter

The second application is a second order biquad lowpass filter circuit, shown in Fig. 2. This example is taken from [31]. Recently, analog and mixed signal circuit design has attracted a lot of attention of researchers in microelectronic systems design. With this example, we attemped to show the applicability of hybrid systems techniques to formal verification of such circuits.
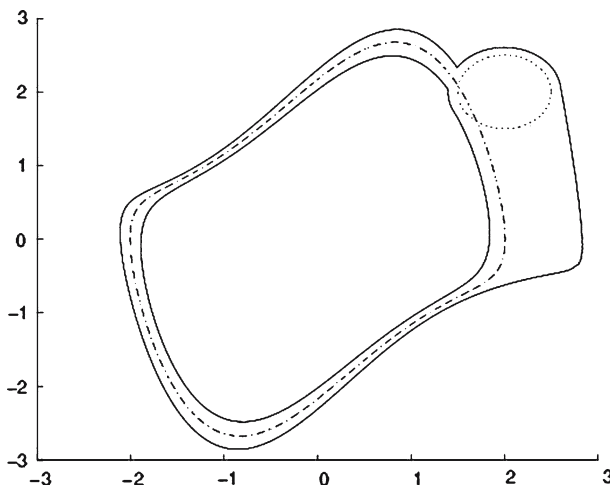

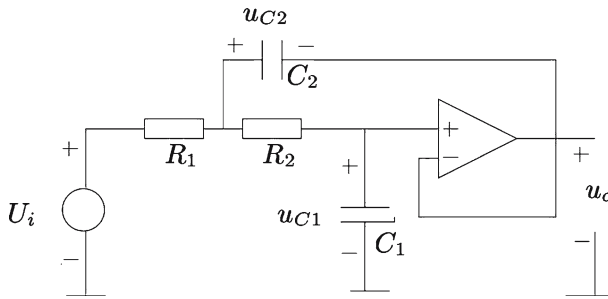**Fig. 1** Reachable set of the Van der Pol oscillator

Fig. 2 Lowpass filter

The behavior of the circuit can be described by the following differential-algebraic equations:

$$\dot{u}_{C1} = 7\frac{u_{C2} + u_o - u_{C1}}{C_1 R_2}, \tag{11}$$

$$\dot{u}_{C2} = \frac{U_i - u_{C2} - u_o}{C_2 R_1} - \frac{u_{C2} + u_o - u_{C1}}{C_2 R_2}, \tag{12}$$

$$u_o - V_{\max} \tanh\left(\frac{(u_{C2} - u_o)V_e}{V_{\max}}\right) + U_{\mathrm{om}} = 0, \tag{13}$$
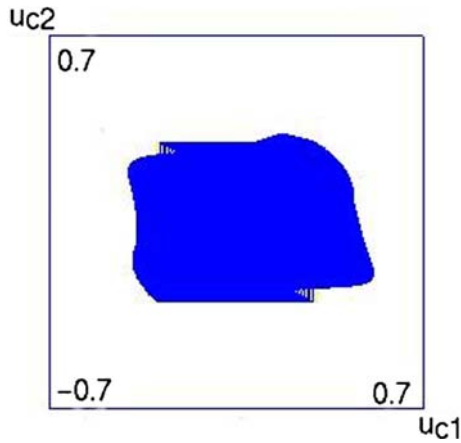
$$i_o = -C_2 \dot{u}_{C2}, \tag{14}$$

$$U_{\mathrm{om}} = \mathcal{V}(i_0), \tag{15}$$

$$\mathcal{V}(i_o) = K i_o + 0.5\sqrt{K_1 i_o^2 - 2K_1 i_o I_s + K_1 I_s^2 + K_2}$$
$$-0.5\sqrt{K_1 i_o^2 + 2K_1 i_o I_s + K_1 I_s^2 + K_2}. \tag{16}$$

The state variables are (u_{C1}, u_{C2}), the voltages across the capacitors C_1 and C_2. The algebraic constraints (13–16) come from the characteristics of the operational amplifier (OPAM) where u_o is the output voltage and U_{om} is to the output voltage decrease caused by the output current i_o. The other variables are circuit parameters. By differentiating (13) the circuit equations can be transformed into a nonlinear ODE on a manifold as with state variables $x = (u_{C1}, u_{C2}, u_o)$. Then, the resulting system is treated using the method for ODEs on manifolds [18]. This method indeed combines the hybridization method to deal with the differential part of the dynamics and the projection integration approach to deal with the algebraic part.

A property of interest to verify is the absence of overshoots. The parameters of the circuit equations are: $K = 0.1e11$, $K_1 = 0.1e21$, $K_2 = 2e4$, $V_e = 1e4$, $V_{\max} = 1.5$, $I_s = 0.5e - 2$. For the highly damped case (where $C_1 = 0.5e - 8$, $C_2 = 2e - 8$, and $R_1 = R_2 = 1e6$), Figure 3 shows the projection of the reachable set on u_{C1} and u_{C2}. The reachable set here is represented as a set of convex polyhedra. The hybridization was done using a mesh of size $h = 0.1$. The initial set is a box: $u_{C1} \in [-0.3, 0.3]$, $u_{C2} \in [-0.3, 0.3]$ and $u_o \in [-0.2, 0.2]$. From the figure, one can see that u_{C1} indeed remains in the range $[-2, 2]$, as desired. Details on the computation results can be found in [18].

Fig. 3 Reachable set projected on variables u_{C1} and u_{C2}

7 Concluding remarks

In this paper, we proposed a framework for approximate analysis of complex non-linear systems by means of approximate systems that we call hybridizations. We also developed two methods for constructing hybridizations, which allows analyzing the original system with an arbitrary precision and a good convergence rate. These results can be readily applied to the verification of interesting properties of hybrid systems.

The results presented in the paper open various interesting directions for future research. One direction concerns the problem of hierarchical mesh refinement that is guided by the information obtained in the process of proving the property, as in the abstraction approaches. Another promising direction is to use mixed rectangular-simplicial meshes in order to achieve a good trade-off between accuracy and computational cost. In addition, the convergence can be improved by using higher degree approximants, such as piecewise quadratic, and the reachability method for polynomial systems [17] can then be used. Finally, an important theoretical question to address is whether other new properties can be verified using the hybridization approach.

Appendix

Proof of Proposition 1

Proposition 4 *If f is λ-Lipschitz, then*

$$\varepsilon(h) \leq h\frac{2n\,\lambda}{n+1} = \overline{\varepsilon}(h).$$

Proof We first estimate an upper bound of $||f(x) - f_l(x)||$ for all points x inside a mesh element $\mathcal{M}_l$. Let v be a vertex of $\mathcal{M}_l$. By the triangle inequality, we have $||f(x) - f_l(x)|| \leq ||f(x) - f(v)|| + ||f(v) - f_l(x)||$. The function f is λ-Lipschitz, then

$$||f(x) - f_l(x)|| \leq \lambda ||x - v|| + ||f(v) - f_l(x)||. \tag{17}$$

Let $V(\mathcal{M}_l) = \{v_0, v_1, \ldots, v_n\}$. A point $x \in V(\mathcal{M}_l)$ can be written as:

$$x = \sum_{i=0}^{n} \alpha_i v_i, \quad \text{with} \sum_{i=0}^{n} \alpha_i = 1 \quad \text{and} \quad \forall i \in \{1, \ldots, n\}, \alpha_i \geq 0. \tag{18}$$

Since f_l is affine, we have

$$f_l(x) = \sum_{i=0}^{n} \alpha_i f_l(v_i) = \sum_{i=0}^{n} \alpha_i f(v_i).$$

Thus,

$$||f(v) - f_l(x)|| \leq ||f(v) - \sum_{i=0}^{n} \alpha_i f(v_i)|| \leq \sum_{i=0}^{n} \alpha_i ||f(v) - f(v_i)||$$

$$\leq \lambda \sum_{i=0}^{n} \alpha_i ||v - v_i||.$$

Equation (17) becomes

$$||f(x) - f_l(x)|| \leq 2\lambda \sum_{i=0}^{n} \alpha_i ||v - v_i||.$$

Note that the above inequality holds for any vertex $v \in V(\mathcal{M}_l)$. We observe, from the conditions (18), that there exists $j \in \{0, \ldots, n\}$ such that $\alpha_j \geq \frac{1}{n+1}$. Since for all $i \neq j$, $||v_i - v_j|| \leq h$ then

$$||f(x) - f_l(x)|| \leq 2\lambda \sum_{i=0}^{n} \alpha_i ||v_j - v_i|| \leq 2\lambda(1 - \alpha_j)h \leq h\frac{2n\lambda}{n+1}.$$

This completes the proof of the proposition. $\square$

An important remark is that the second partial derivatives of the linear approximation f_l vanish; therefore, if f is a C^2 function, we can obtain a better error bound. Exceptionally, to write the second partial derivatives of f, we use superscripts to indicate the components of f, that is $f = (f^1, f^2, \ldots, f^n)$.

Proof of Proposition 2

Proposition 5 *If f is C^2 on Ω with bounded second order derivatives then*

$$\varepsilon(h) \leq \frac{Kn^2}{2(n+1)^2}h^2 = \bar{\varepsilon}(h)$$

where

$$K = \max_{i \in \{1, \ldots, n\}} \sup_{x \in \Omega} \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \left| \frac{\partial^2 f^i(x)}{\partial x_{p_1} \partial x_{p_2}} \right|.$$

Proof For a given mesh element $\mathcal{M}_l$, we define the function $e(x) = f^i(x) - f^i_l(x)$ where f^i and f^i_l denotes the i-th components of the vector fields f and f_l. Let $x^* = \arg\max_{x \in \mathcal{M}_l} |e(x)|$ (note that the simplex $\mathcal{M}_l$ is compact). Let v be a vertex of $\mathcal{M}_l$, and all points in the line segment connecting x^* and v can be written as: $x(\gamma) = x^* + \gamma (v - x^*)$, $\gamma \in [0, 1]$. To determine a bound on $e(x^*)$, we define a function $z(\gamma) = e(x(\gamma))$ for $\gamma \in [0, 1]$. Expanding z with respect to γ gives

$$z(1) = z(0) + \frac{dz}{d\gamma}(0) + \int_0^1 \frac{d^2z}{d\gamma^2}(s)\,(1-s)ds. \tag{19}$$

We can see that $dx(\gamma)/d\gamma = (v - x^*)$. Thus,

$$\frac{dz}{d\gamma}(\gamma) = \sum_{p_1=1}^{p_1=n} \frac{\partial e}{\partial x_{p_1}}(x(\gamma))\,(v_{p_1} - x^*_{p_1}),$$

$$\frac{d^2z}{d\gamma^2}(\gamma) = \sum_{p_1=1}^{p_1=n}\sum_{p_2=1}^{p_2=n} \frac{\partial^2 e}{\partial x_{p_1} \partial x_{p_2}}(x(\gamma))\,(v_{p_1} - x^*_{p_1})\,(v_{p_2} - x^*_{p_2}).$$

Since the second order derivatives of f_l vanish, then

$$\forall p_1, p_2 \in \{1, \dots, n\}, \ \frac{\partial^2 e}{\partial x_{p_1} \partial x_{p_2}} = \frac{\partial^2 f^i}{\partial x_{p_1} \partial x_{p_2}}.$$

Similar to the proof of Lemma 1, we can show that there exists $v \in V(\mathcal{M}_l)$ such that $||v - x^*|| \le hn/(n+1)$. Then, using the bound K on the second order derivatives of the function f, we obtain

$$\left|\frac{d^2z}{d\gamma^2}(\gamma)\right| \le h^2 \frac{n^2 K}{(n+1)^2}.$$

In addition, $|e(x)|$ attains a maximum at x^*, which implies that $dz(0)/d\gamma = 0$. By definition of the interpolating function, $f^i(v) = f^i_l(v)$, then $z(1) = 0$. Therefore, (19) becomes:

$$f^i(x^*) - f^i_l(x^*) + \int_0^1 \frac{d^2z}{d\gamma^2}(s)\,(1-s)ds = 0.$$

Using the above bound on the second order derivative of $z(\gamma)$, we get

$$|f^i(x^*) - f^i_l(x^*)| \le h^2 \frac{n^2 K}{(n+1)^2} \int_0^1 (1-s)ds = h^2 \frac{n^2 K}{2(n+1)^2}.$$

We complete the proof by remarking that such an equality holds for all components of the vector fields f and f_l. $\qquad\square$

References

1. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. **138**(1), 3–34 (1995)
2. Alur, R., Dang, T., Ivancic, F.: Counter-example guided predicate abstraction of hybrid systems. Theor. Comput. Sci. **354**(2), 250–271 (2006)
3. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994)
4. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.: Discrete abstractions of hybrid systems. Proc. IEEE **88**(2), 971–984 (2000)
5. Anai, H., Weispfenning, V.: Reach set computations using real quantifier elimination. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) Hybrid Systems: Computation and Control, vol. 2034 in LNCS, pp. 63–75. Springer, Heidelberg (2001)
6. Asarin, E., Bournez, O., Dang, T., Maler, O.: Approximate reachability analysis of piecewise-linear dynamical systems. In: Krogh, B.H., Lynch, N. (eds.) Hybrid Systems: Computation and Control, vol. 1790 in LNCS, pp. 20–31. Springer, Heidelberg (2000)
7. Asarin, E., Dang, T., Maler, O.: d/dt: A tool for verification of hybrid systems. In: Ed Brinksma, Kim Guldstrand Larsen (eds.) Computer Aided Verification, vol. 2404 in LNCS, pp. 365–370. Springer, Heidelberg (2002)
8. Asarin, E., Dang, T.: Abstraction by projection and application to multi-affine systems. In: Alur, R., Pappas, G.J. (eds) Hybrid Systems: Control and Computation, vol. 2993 LNCS, pp. 32–47. Springer, Heidelberg (2004)
9. Asarin, E., Dang, T., Girard, A.: Reachability analysis of nonlinear systems using conservative approximations. In: Maler, O., Pnueli, A. (eds.) Hybrid Systems: Computation and Control, vol. 2623 in LNCS, pp. 20–35. Springer, Heidelberg (2003)
10. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. Theor. Comput. Sci. **138**(1), 35–66 (1995)
11. Aubin, J.-P., Lygeros, J., Quincampoix, M., Sastry, S., Seube, N.: Impulse differential inclusions: a viability approach to hybrid systems. IEEE Trans. Autom. Control, **47**(1), 2–20 (2002)
12. Barringer, H., Kuiper, R., Pnueli, A.: A really abstract concurrent model and its temporal logic. In: POPL'86: Principles of Programming Languages, pp. 173–183 (1986)
13. Botchkarev, O., Tripakis, S.: Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In: Krogh B., Lynch N. (eds.), Hybrid Systems: Computation and Control, vol. 1790 in LNCS, pp. 73–88. Springer, Heidelberg (2000)
14. Branicky, M.S., Borkar, V.S., Mitter, S.K.: A unified framework for hybrid control: model and optimal control theory. IEEE Trans. on Automatic Control, **43**(1), 31–45 (1998)
15. Chutinan, A., Krogh, B.H.: Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In: Vaandrager, F.W., van Schuppen, J.H. (eds.) Hybrid systems: Computation and Control, vol. 1569 in LNCS, pp. 76–90. Springer, Heidelberg (1999)
16. Clarke, E., Fehnker, A., Han, Z., Krogh, B., Ouaknine, J., Stursberg, O., Theobald, M.: Abstraction and counterexample-guided refinement in model checking of hybrid systems. Int. J. Foundations of Comput. Sci. **14**(4), 583–604 (2003)
17. Dang, T.: Approximate reachability computation for polynomial systems. In: Hespanha, J., Tiwari, A. (eds.) Hybrid Systems: Control and Computation, vol. 3927 in LNCS, pp. 138–152. Springer, Heidelberg (2006)
18. Dang, T., Donze, A., Maler, O.: Verification of analog and mixed-signal circuits using hybrid systems techniques. In: Hu, A., Martin, A. (eds.) Formal Methods for Computer Aided Design, vol. 3312 in LNCS, pp. 21–36. Springer, Heidelberg (2004)
19. Dang, T., Maler, O.: Reachability Analysis via Face Lifting. In: Henzinger, T.A., Sastry, S. (eds.), Hybrid Systems: Computation and Control, vol. 1386 in LNCS, pp. 96–109. Springer, Heidelberg (1998)
20. Davoren, J.M., Coulthard, V., Markey, N., Moor, T.: Non-deterministic temporal logics for general flow systems. In: Alur, R., G.J. Pappas, G.J. (eds.) Hybrid Systems: Computation and Control HSCC04, vol. 2993 in LNCS, pp. 280–295. Springer, Heidelberg (2004)
21. Decarlo, R.A., Branicky, M.S., Pettersson, S., Lennartson, B.: Perspectives and results on the stability and stabilizability of hybrid systems. Proc. the IEEE, **88**(7), 1069–1082 (2000)
22. Della Dora, J., Maignan, A., Mirica-Ruse, M., Yovine, S.: Hybrid computation. In: Proceedings International Symposium on Symbolic and Algebraic Computation ISSAC'01 (2001)
23. Dieudonné, J.: Calcul Infinitésimal. Collection Méthodes. Hermann, Paris (1968)

24. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, pp. 995–1072. Elsevier, Amsterdam (1990)
25. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech. In: Morari, M., Thiele, L. (eds.) Hybrid Systems: Computation and Control, vol. 3414 in LNCS, pp. 258–273. Springer, Heidelberg (2005)
26. Girard, A.: Approximate solutions of ODEs using piecewise linear vector fields. In: Proceedings of the Int. Workshop on Computer Algebra in Scientific Computing CASC'02 (2002)
27. Girard, A.: Reachability of uncertain linear systems using zonotopes. In: Morari, M., Thiele, L. (eds.) Hybrid Systems: Computation and Control, vol. 3414 in LNCS, pp. 291–305. Springer, Heidelberg (2005)
28. Greenstreet, M.R., Mitchell, I.: Reachability analysis using polygonal projections. In: Vaandrager, F., van Schuppen, J.H. (eds.) Hybrid Systems: Computation and Control, vol. 1569 in LNCS, pp. 76–90. Springer, Heidelberg (1999)
29. Habets, L.C.G.J.M., van Schuppen, J.H.: Control of piecewise-linear hybrid systems on simplices and rectangles. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) Hybrid systems: Computation and Control, vol.. 2034 LNCS, pp. 261–273. Springer, Heidelberg (2001)
30. Hämmerlin, G., Karl-Heinz Hoffmann, K.-H.: Numerical Mathematics. Springer, Heidelberg (1991)
31. Hartong, W., Hedrich, L., Barke, E.: On discrete modelling and model checking for nonlinear analog systems. In: Ed Brinksma, Kim Guldstrand Larsen (eds.) Computer Aided Verification, vol. 2404 in LNCS, pp. 401–413. Springer, Heidelberg (2002)
32. Henzinger, T.A., Ho, P.-H., Wong-Toi, H.: HyTech: A model checker for hybrid systems. Softw. Tools Technol. Transf. **1**(1–2), 110–122 (1997)
33. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. Syst. Sci. **57**(1), 94–124 (1998)
34. Hubbard, J., West, B.: Differential equations: a dynamical system approach, part 2: higher dimensional systems. Texts in Applied Mathematics, **18** Springer, Heidelberg (1995)
35. Johansson, M., Rantzer, A.: Computation of piecewise quadratic Lyapunov functions for hybrid systems. IEEE Trans. Autom. Control **43**(4), 555–559 (1998)
36. Kloetzer, M., Belta, C.: Reachability analysis of multi-affine systems. In: Hespanha, J., Tiwari, A. (eds.) Hybrid Systems: Computation and Control, vol. 3927 in LNCS, pp. 348–362. Springer, Heidelberg (2006)
37. Kratz, F., Sokolsky, O., Pappas, G.J., Lee, I.: R-Charon : a modeling language for reconfigurable hybrid systems. In: Hespanha, J., Tiwari, A. (eds.) Hybrid Systems: Computation and Control, vol. 3927 in LNCS, pp. 392–406. Springer, Heidelberg (2006)
38. Kuhn, H.W.: Some combinatorial lemmas in topology. IBM J. Res. Dev. **4**(5), 518–524 (1960)
39. Kurzhanski, A., Varaiya, P.: Ellipsoidal techniques for reachability analysis. In: Krogh, B., Lynch, N. (eds.) Hybrid Systems: Computation and Control, vol. 1790 in LNCS, pp. 202–214. Springer, Heidelberg (2000)
40. Lafferriere, G., Pappas, G., Yovine, S.: Symbolic reachability computation of families of linear vector fields. J. Symbolic Comput. **32**(3), 231–253 (2001)
41. Larsen, K., Pettersson, P., Yi, W.: Uppaal in a nutshell. Soft. Tools Technol. Transf. **1**(1–2), 134–152 (1997)
42. Liberzon, D.: Switching in systems and control. Volume in series Systems and Control: Foundations and Applications. Birkhauser, Boston (2003)
43. Manna, Z., Pnueli, A.: The temporal logic of reactive and concurrent systems: specification. Springer, New York (1991)
44. Mitchell, I., Templeton, J.A.: A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid Systems. In: Morari, M., Thiele, L. (eds.) Hybrid Systems: Computation and Control, vol. 3414 in LNCS, pp. 480–494. Springer, Heidelberg (2005)
45. Kvasnica, M., Grieder, P., Baoti, M., Morari, M.: Multi-Parametric Toolbox (MPT). In: Alur, R., Pappas, G.J. (eds.) Hybrid Systems: Computation and Control, vol. 2993 in LNCS, pp. 448–462. Springer, Heidelberg (2004)
46. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Alur, A., Pappas, G. (eds.) Hybrid Systems: Computation and Control, vol. 2993 LNCS, pp. 477–492. Springer, Heidelberg (2004)
47. Prajna, S., Papachristodoulou, A.: Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In: Proceedings of the American Control Conference ACC (2003)

48. Puri, A., Varaiya, P.: Verification of sybrid systems using abstraction. In: Antsaklis, P., Kohn, W., Nerode, A., Sastry, S. (eds.) Hybrid Systems II, vol. 999 in LNCS, pp. 359–369, Springer, Heidelberg (1995)
49. Rondepierre, A., Dumas, J.G.: Algorithms for hybrid optimal control. Technical report IMAG-ccsd-00004191, arXiv math.OC/0502172 (2005)
50. Saint-Pierre, P.: Approximation of Viability Kernels and Capture Basin for Hybrid Systems. In: Proceedings of European Control Conference ECC'01, pp. 2776–2783 (2001)
51. Sastry, S.: Nonlinear systems: analysis, stability and control. Springer, Heidelberg (1999)
52. Stuart, A.M., Humphries, A.R.: Dynamical systems and numerical analysis. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (1996)
53. Stursberg, O., Kowalewski, S.: Approximating switched continuous systems by rectangular automata. In: Proceedings European Control Conference ECC, (1999)
54. Tabuada, P., Pappas, G.: Model-checking LTL over controllable linear systems is decidable, In: Maler, O., Pnueli, A. (eds.) Hybrid Systems: Computation and Control, vol. 2623 LNCS, pp. 498–513, Springer, Heidelberg (2003)
55. Tiwari, A., Khanna, G.: Series of abstractions for hybrid automata. In: Tomlin, C., Greenstreet, M.R. (eds.) Hybrid Systems: Computation and Control, vol. 2289 LNCS, pp. 465–478, (2002)
56. Tiwari, A., Khanna, A.: Nonlinear systems: approximating reach sets. In: Alur, R., Pappas, G.J. (eds.) Hybrid Systems: Computation and Control, vol. 2993 LNCS, pp. 600–614. Springer, Heidelberg (2004)
57. Tomlin, C., Mitchell, I., Bayen, A., Oishi, M.: Computational techniques for the verification of hybrid systems. Proc. IEEE **91**(7), 986–1001 (2003)
58. Torrisi, F.D., Bemporad, A.: HYSDEL - A tool for generating computational hybrid models. IEEE Trans. Control Syst. Technol. **12**(2), 235–249 (2004)
59. Van der Schaft, A.J., Schumacher, J.M.: An introduction to hybrid dynamical systems. Lect. Notes in Control and Information Sciences, Vol 251. Springer, London, (2000)
60. Yovine, S.: Kronos: A verification tool for real-time systems. Soft. Tools Technol. Trans. **1**(1–2), 123–133 (1997)